

H5PL_SET_LOADING_STATE

[Expand all](#) [Collapse all](#)

- [Jump to ...](#)
- [Summary](#)
- [Description](#)
- [Example](#)
- [Switch language ...](#)
- [C](#)
- [C++](#)
- [FORTRAN](#)
- [JAVA](#)

[Summary](#)
[Description](#)
[Example](#)
[JAVA](#)
[FORTRAN](#)
[C++](#)
[C](#)

H5PL_SET_LOADING_STATE

Controls the loading of dynamic plugin types

Procedure:

H5PL_SET_LOADING_STATE (plugin_control_mask)

Signature:

```
herr_t H5PLset_loading_state( unsigned int plugin_control_mask )
```

Parameters:

unsigned int plugin_control_mask

IN: The list of dynamic plugin types to enable or disable.

A plugin bit set to 0 (zero) prevents use of that dynamic plugin.

A plugin bit set to 1 (one) enables use of that dynamic plugin.

Setting `plugin_control_mask` to a negative value enables all dynamic plugins.

Setting `plugin_control_mask` to 0 (zero) disables all dynamic plugins.

Description:

Motivation:

The loading of external dynamic plugins can be controlled during runtime with an environment variable, `HDF5_PLUGIN_PRELOAD`.

The environment variable can control the loading of dynamic filters at runtime, but it will disable it for all running programs that access that variable using the library.

H5PL_SET_LOADING_STATE controls the loading of external dynamic plugins during program execution.

H5PL_SET_LOADING_STATE uses one argument to enable or disable individual plugins.

The `plugin_control_mask` parameter is an encoded integer in which each bit controls a specific plugin or class of plugins. Bit positions allocated to date are listed in the "Plugin Type/Bit Position" table below.

A plugin bit set to 0 (zero) prevents the use of the dynamic plugin corresponding to that bit position. A plugin bit set to 1 (one) allows the use of that dynamic plugin.

All dynamic plugins can be enabled by setting `plugin_control_mask` to a negative value. A value of 0 (zero) will disable all dynamic plugins.

H5PL_SET_LOADING_STATE inspects the `HDF5_PLUGIN_PRELOAD` environment variable every time it is called. If the environment variable is set to the special `::` string, all dynamic plugins will be disabled.

The classes of plugins subject to programmatic control are specified in the `H5PL_type_t` *ENUM*. Since only dynamically loaded filter plugins are currently subject to programmatic control, only one plugin type is defined:

Plugin Type	Bit Position
<code>H5PL_TYPE_FILTER</code>	0

Additional values may be added to the `H5PL_type_t` *ENUM* as programmatic control is implemented for other plugin types.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

Example:

```
/* Disable plugin X: Requires user to negate the state
 * with a 0 in bit position X and AND it with the result
 * from an H5PLget_loading_state call. */

H5PLget_loading_state(&curr_setting);
int new_setting = curr_setting & ~H5PL_FILTER_PLUGIN ;
H5PLset_loading_state (new_setting);
```

History:

Release	Change
1.8.15	Function introduced in this release.